PL2303HXD Android Development Package

Prolific
TECHNOLOGY FOR TOMORROW

Agenda

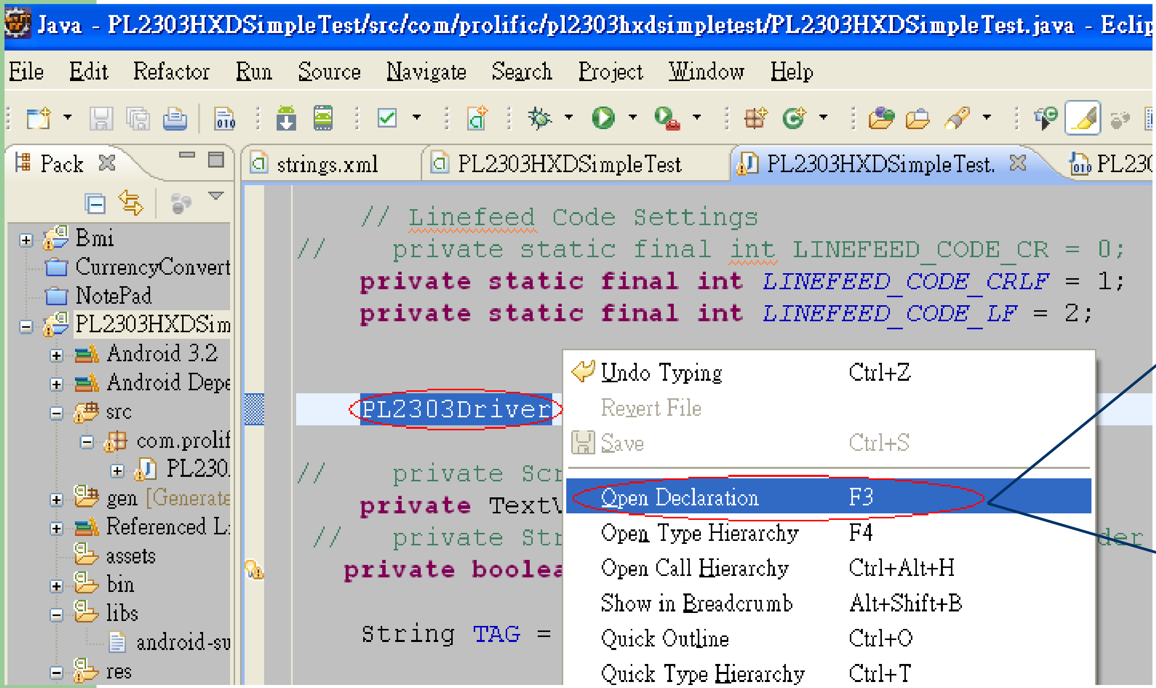
- PL2303Driver Class Declaration
- Documents Index
- PL2303Driver Class
- PL2303Driver.BaudRate Class
- PL2303Driver.DataBits Class
- PL2303Driver.FlowControl Class
- PL2303Driver.Parity Class
- PL2303Driver.StopBits Class

Home

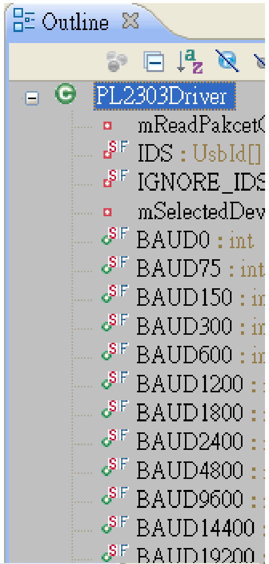
- Implementing USB Interface in Android
  - Dynamically registered BroadcastReceiver
  - PendingIntent
  - Read Data
  - Write Data
- The Drop-Down Menu Program

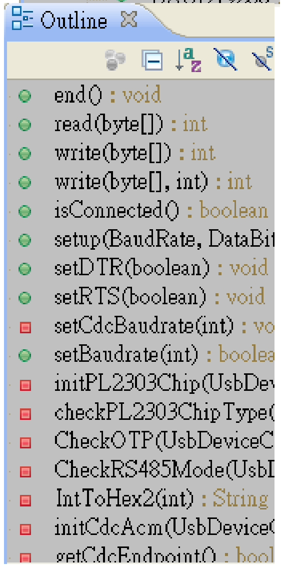
Home

PL2303Driver Class Declaration



```
    // Linefeed Code Settings
//    private static final int LINEFEED_CODE_CR = 0;
    private static final int LINEFEED_CODE_CRLF = 1;
    private static final int LINEFEED_CODE_LF = 2;


    PL2303Driver

//    private Scr
    private TextV
 //   private Str                              der
   private boolea

    String TAG =
```



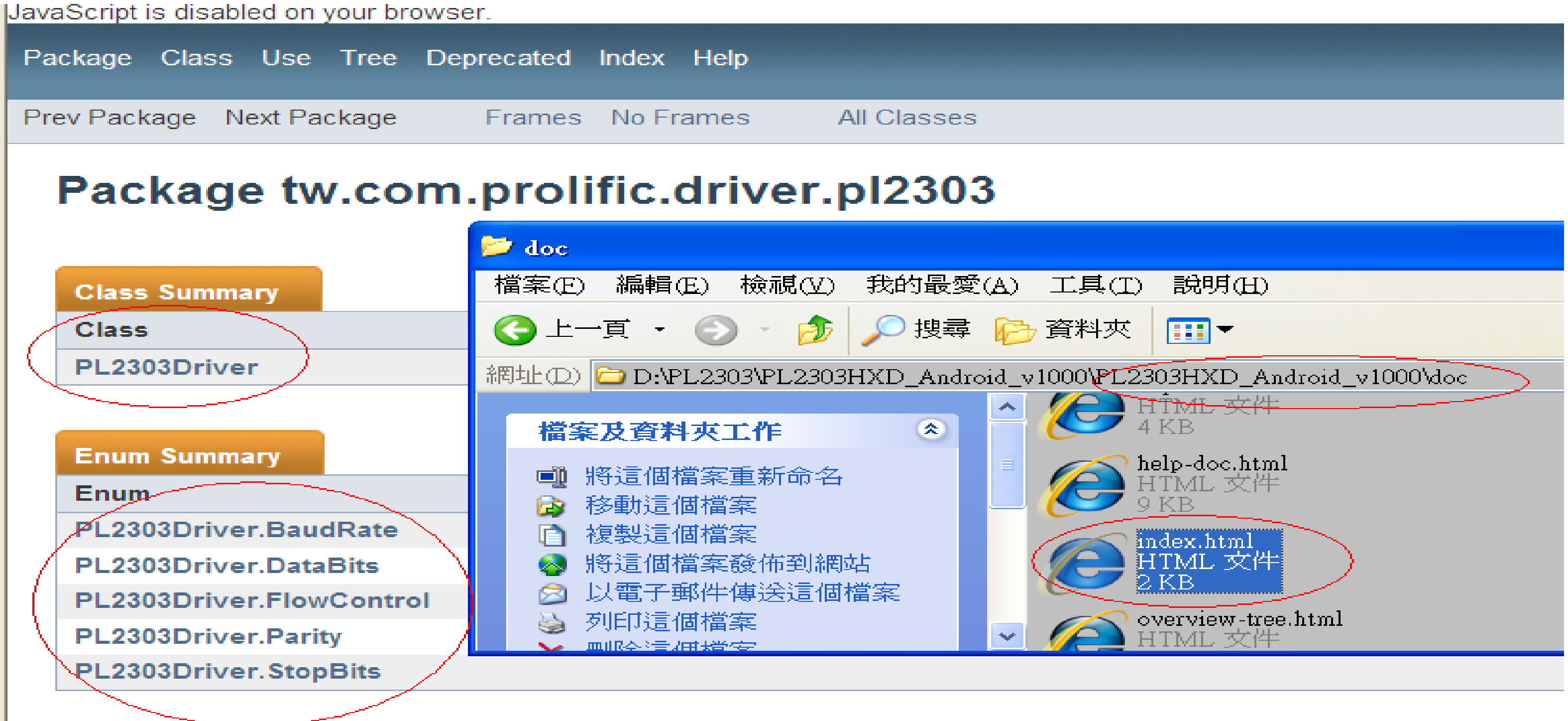


Home

Documents Index

- PL2303HXD_Android_v1000\doc\Index.html

To develop Android AP need to refer to the doc files, the start file as index.html



JavaScript is disabled on your browser.

Package tw.com.prolific.driver.pl2303

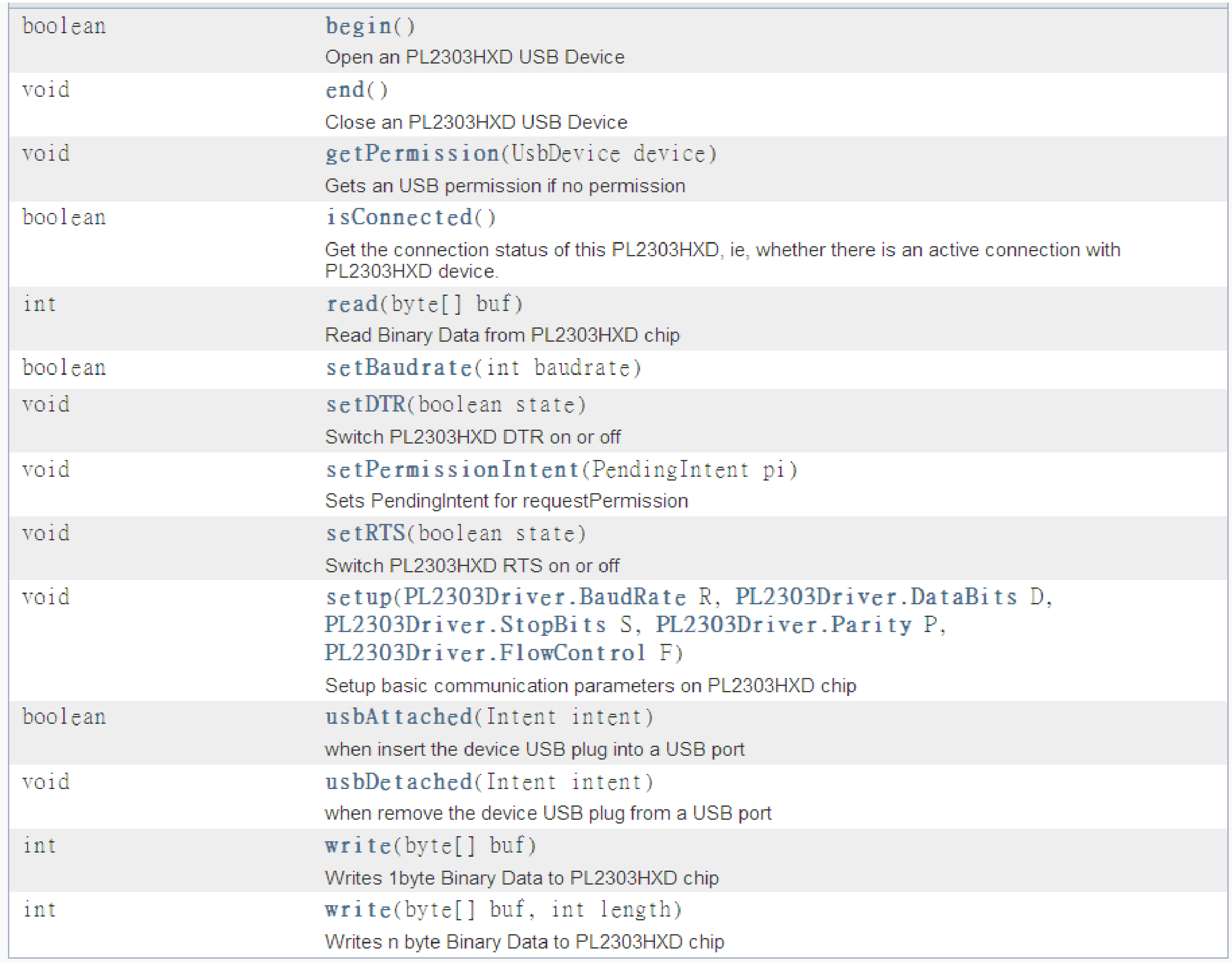
Class Summary
Class
PL2303Driver

Enum Summary
Enum
PL2303Driver.BaudRate
PL2303Driver.DataBits
PL2303Driver.FlowControl
PL2303Driver.Parity
PL2303Driver.StopBits

網址(D) D:\PL2303\PL2303HXD_Android_v1000\PL2303HXD_Android_v1000\doc

Home

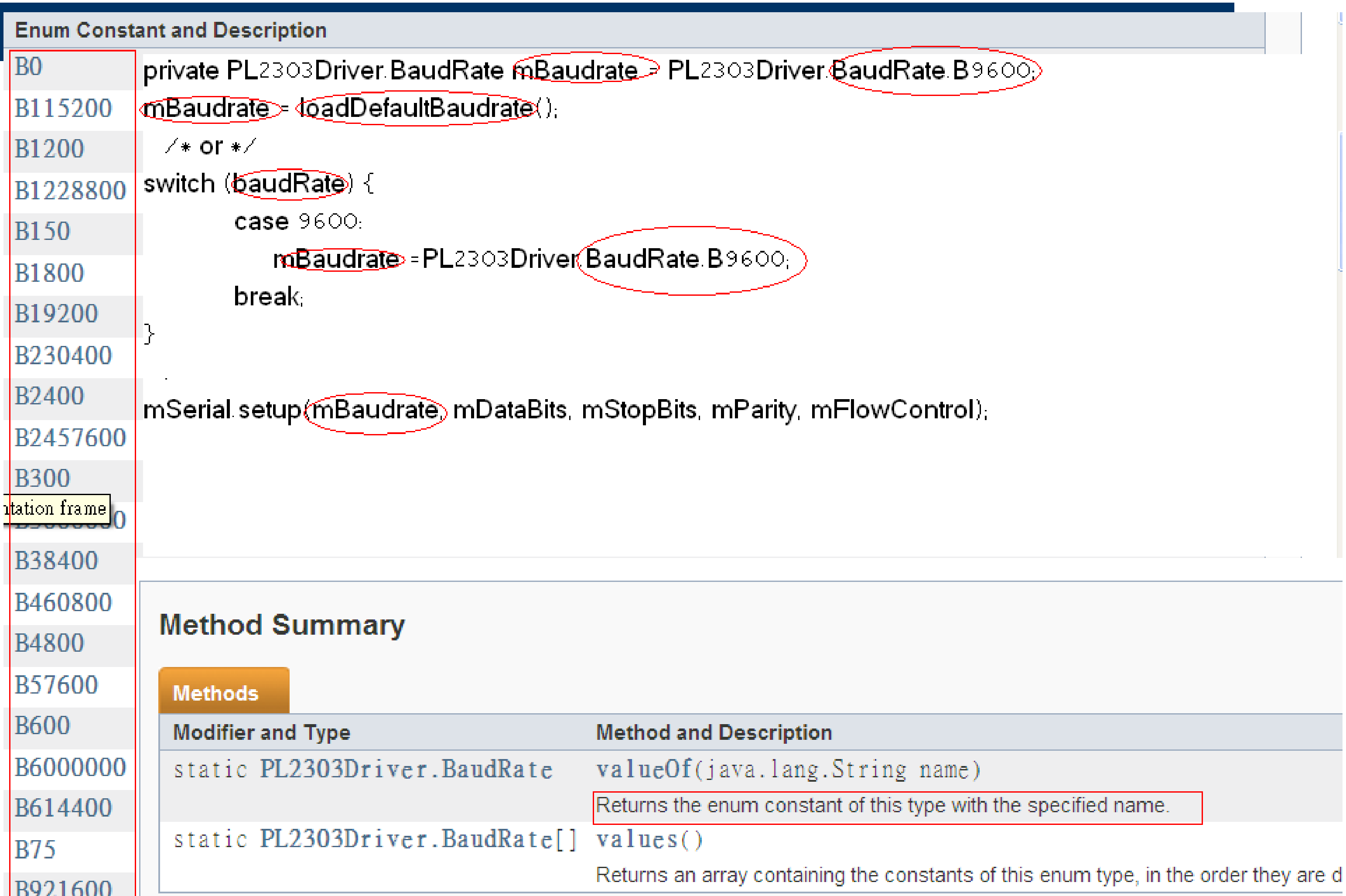
PL2303Driver Class



boolean	**begin**() Open an PL2303HXD USB Device
void	**end**() Close an PL2303HXD USB Device
void	**getPermission**(UsbDevice device) Gets an USB permission if no permission
boolean	**isConnected**() Get the connection status of this PL2303HXD, ie, whether there is an active connection with PL2303HXD device.
int	**read**(byte[] buf) Read Binary Data from PL2303HXD chip
boolean	**setBaudrate**(int baudrate)
void	**setDTR**(boolean state) Switch PL2303HXD DTR on or off
void	**setPermissionIntent**(PendingIntent pi) Sets PendingIntent for requestPermission
void	**setRTS**(boolean state) Switch PL2303HXD RTS on or off
void	**setup**(PL2303Driver.BaudRate R, PL2303Driver.DataBits D, PL2303Driver.StopBits S, PL2303Driver.Parity P, PL2303Driver.FlowControl F) Setup basic communication parameters on PL2303HXD chip
boolean	**usbAttached**(Intent intent) when insert the device USB plug into a USB port
void	**usbDetached**(Intent intent) when remove the device USB plug from a USB port
int	**write**(byte[] buf) Writes 1byte Binary Data to PL2303HXD chip
int	**write**(byte[] buf, int length) Writes n byte Binary Data to PL2303HXD chip

Home

PL2303Driver.BaudRate Class



Enum Constant and Description

B0
B115200
B1200
B1228800
B150
B1800
B19200
B230400
B2400
B2457600
B300
tation frame
0
B38400
B460800
B4800
B57600
B600
B6000000
B614400
B75
B921600

```
private PL2303Driver.BaudRate mBaudrate = PL2303Driver.BaudRate.B9600;
mBaudrate = loadDefaultBaudrate();
  /* or */
switch (baudRate) {
        case 9600:
            mBaudrate = PL2303Driver.BaudRate.B9600;
            break;
}

mSerial.setup(mBaudrate, mDataBits, mStopBits, mParity, mFlowControl);
```

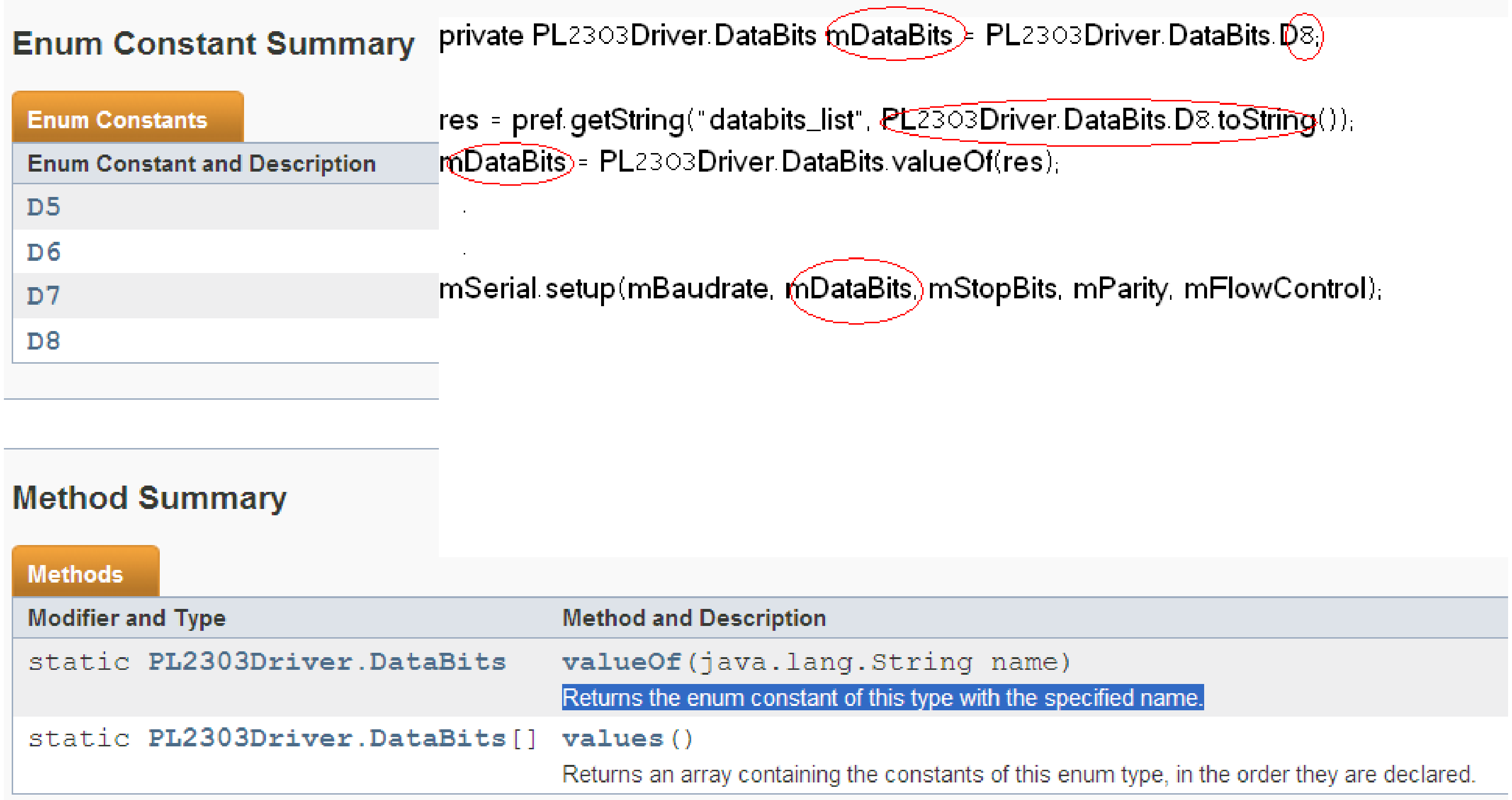
Method Summary

Methods

Modifier and Type	Method and Description
`static PL2303Driver.BaudRate`	`valueOf(java.lang.String name)` Returns the enum constant of this type with the specified name.
`static PL2303Driver.BaudRate[]`	`values()` Returns an array containing the constants of this enum type, in the order they are d

Home

PL2303Driver.DataBits Class



Enum Constant Summary

Enum Constants

Enum Constant and Description
D5
D6
D7
D8

```
private PL2303Driver.DataBits mDataBits = PL2303Driver.DataBits.D8;

res = pref.getString("databits_list", PL2303Driver.DataBits.D8.toString());
mDataBits = PL2303Driver.DataBits.valueOf(res);
  .
  .
mSerial.setup(mBaudrate, mDataBits, mStopBits, mParity, mFlowControl);
```

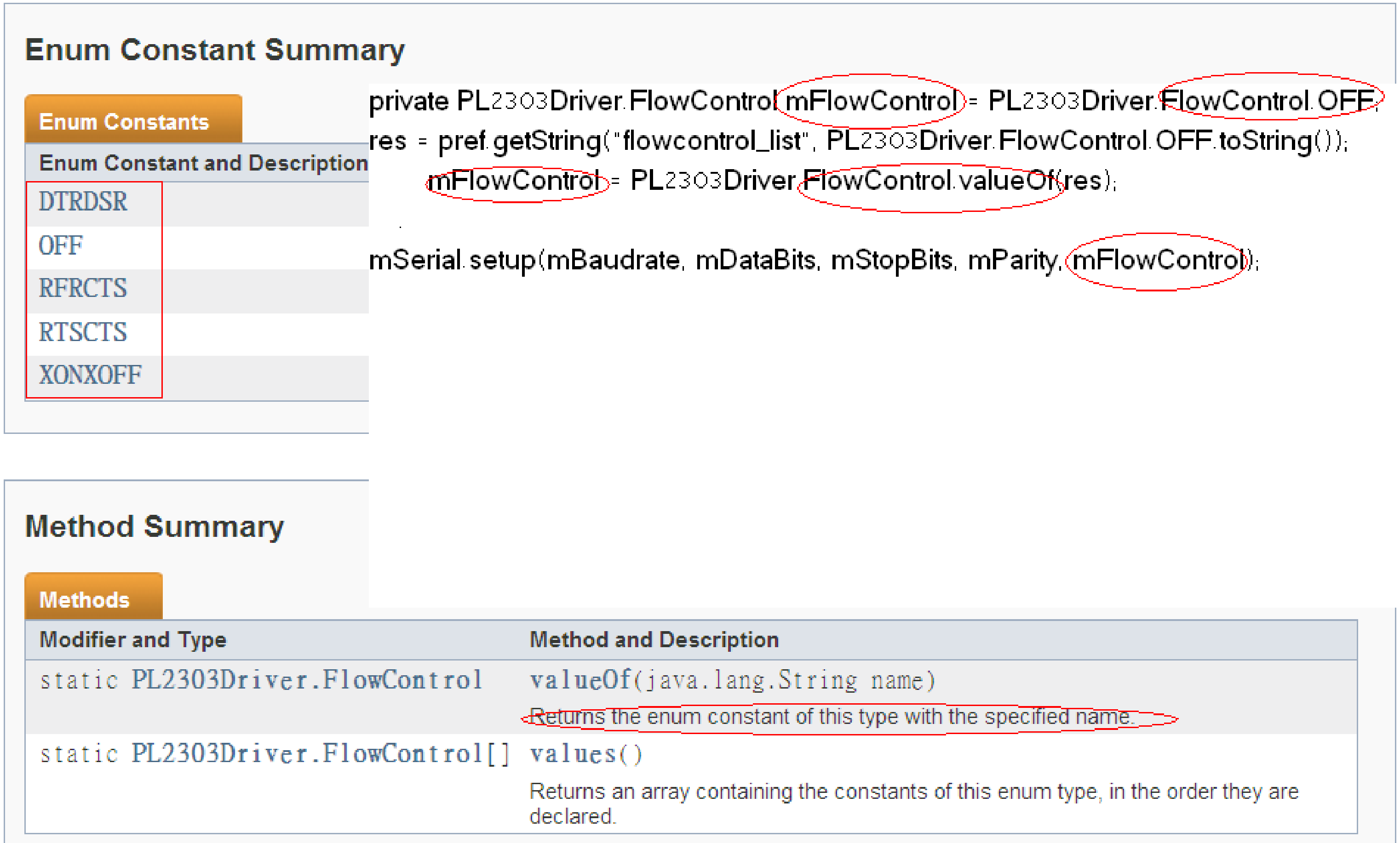
Method Summary

Methods

Modifier and Type	Method and Description
`static PL2303Driver.DataBits`	`valueOf(java.lang.String name)` Returns the enum constant of this type with the specified name.
`static PL2303Driver.DataBits[]`	`values()` Returns an array containing the constants of this enum type, in the order they are declared.

Home

PL2303Driver.FlowControl Class



Enum Constant Summary

Enum Constants

Enum Constant and Description
DTRDSR
OFF
RFRCTS
RTSCTS
XONXOFF

```
private PL2303Driver.FlowControl mFlowControl = PL2303Driver.FlowControl.OFF;
res = pref.getString("flowcontrol_list", PL2303Driver.FlowControl.OFF.toString());
      mFlowControl = PL2303Driver.FlowControl.valueOf(res);

mSerial.setup(mBaudrate, mDataBits, mStopBits, mParity, mFlowControl);
```

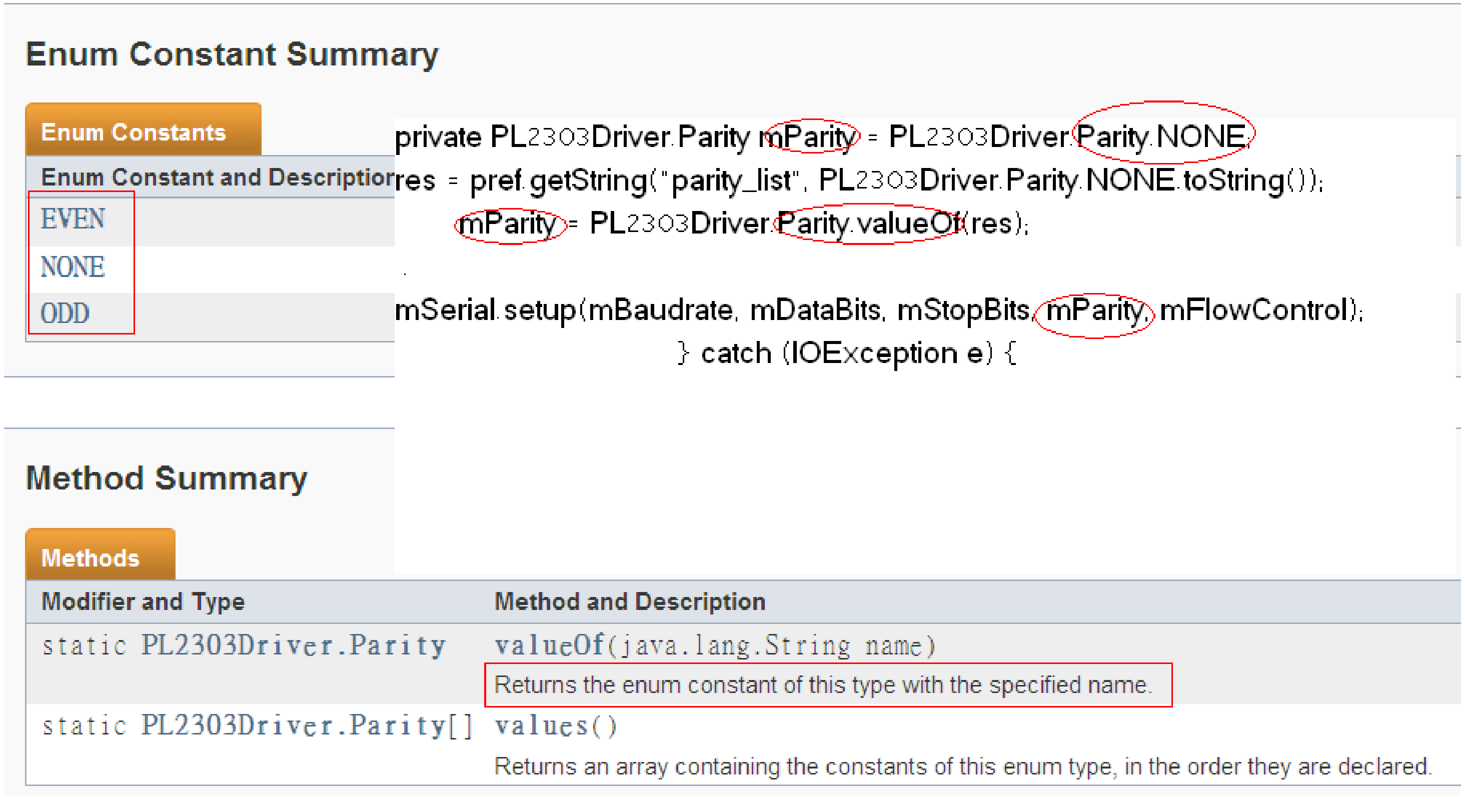
Method Summary

Methods

Modifier and Type	Method and Description
`static PL2303Driver.FlowControl`	`valueOf(java.lang.String name)` Returns the enum constant of this type with the specified name.
`static PL2303Driver.FlowControl[]`	`values()` Returns an array containing the constants of this enum type, in the order they are declared.

Home

PL2303Driver.Parity Class



Enum Constant Summary

Enum Constants

Enum Constant and Descriptio
EVEN
NONE
ODD

```
private PL2303Driver.Parity mParity = PL2303Driver.Parity.NONE;
res = pref.getString("parity_list", PL2303Driver.Parity.NONE.toString());
      mParity = PL2303Driver.Parity.valueOf(res);
.
mSerial.setup(mBaudrate, mDataBits, mStopBits, mParity, mFlowControl);
                              } catch (IOException e) {
```

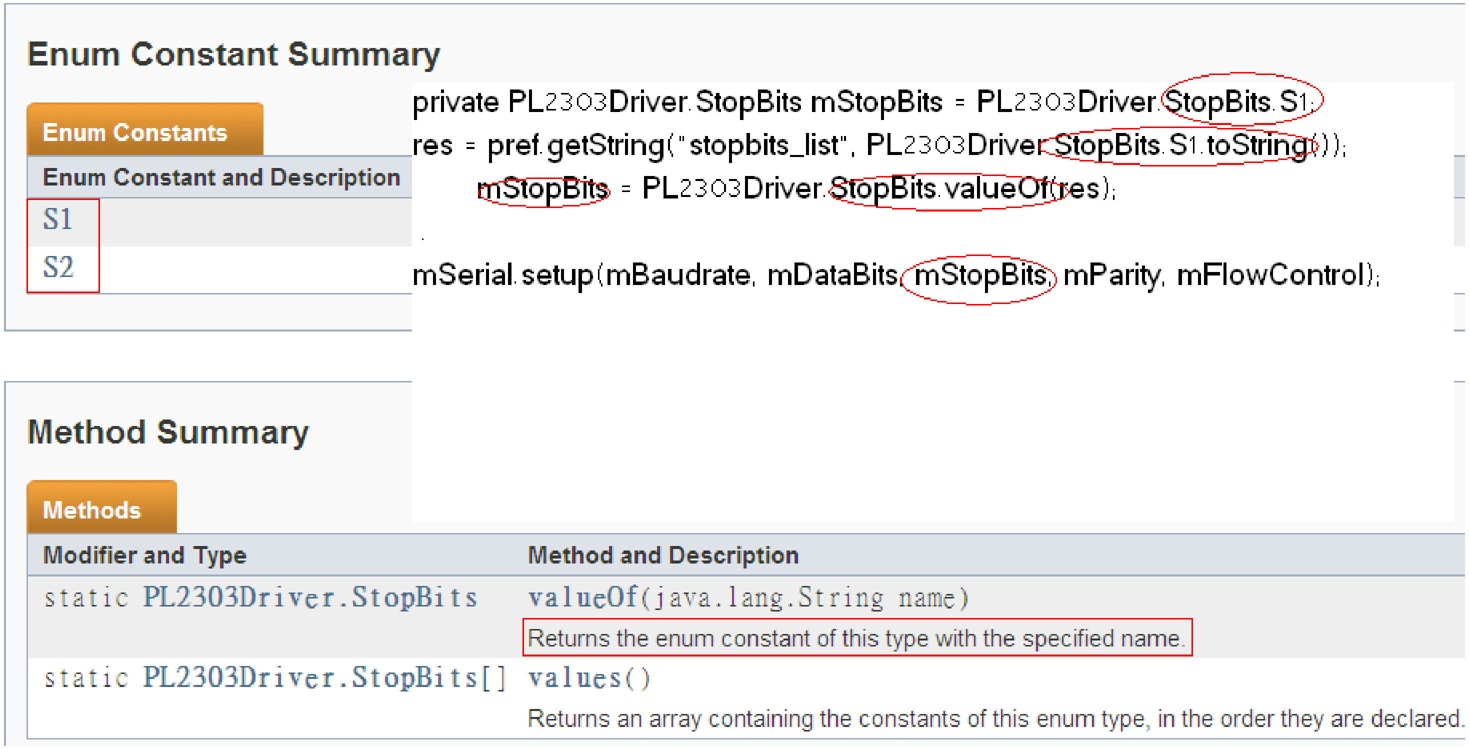
Method Summary

Methods

Modifier and Type	Method and Description
`static PL2303Driver.Parity`	`valueOf(java.lang.String name)` Returns the enum constant of this type with the specified name.
`static PL2303Driver.Parity[]`	`values()` Returns an array containing the constants of this enum type, in the order they are declared.

Home

PL2303Driver.StopBits Class



Enum Constant Summary

Enum Constants

Enum Constant and Description
S1
S2

```
private PL2303Driver.StopBits mStopBits = PL2303Driver.StopBits.S1;
res = pref.getString("stopbits_list", PL2303Driver.StopBits.S1.toString());
     mStopBits = PL2303Driver.StopBits.valueOf(res);
.
mSerial.setup(mBaudrate, mDataBits, mStopBits, mParity, mFlowControl);
```

Method Summary

Methods

Modifier and Type	Method and Description
static PL2303Driver.StopBits	valueOf(java.lang.String name) Returns the enum constant of this type with the specified name.
static PL2303Driver.StopBits[]	values() Returns an array containing the constants of this enum type, in the order they are declared.

Home

Implementing USB Interface in Android

- Implementation of a broadcast receiver (BroadcastReceiver) detect USB interface.

- Android framework layer TelephonyManager underlying remote service tracking, final PendingIntent to track.

- http://torvafirmus-android.blogspot.tw/

- **SMS Example: Use registerReceiver, SmsManager, PendingIntent**

http://shung007.blogspot.tw/2012/04/tqc-android-3-7-use-registerreceiver.html

Home

Dynamically registered BroadcastReceiver

```
/* listen for new devices */
IntentFilter filter = new IntentFilter();

/* BroadcastReceiver specified action, that is to listen for the
message name */
filter.addAction(UsbManager.ACTION_USB_DEVICE_ATTACHED );
filter.addAction(UsbManager.ACTION_USB_DEVICE_DETACHED);

/* Register listener*/
registerReceiver(mUsbReceiver, filter);
```

Home

PendingIntent

```
/* It is the starting point for controlling HW */
mSerial = new PL2303Driver((UsbManager)
getSystemService(Context.USB_SERVICE));

/* Register the intent */
PendingIntent permissionIntent =
PendingIntent.getBroadcast(this, 0, new Intent(
                ACTION_USB_PERMISSION), 0);
mSerial.setPermissionIntent(permissionIntent);
```

Home

```
/* BroadcastReceiver when insert/remove the device USB plug into/from a
USB  port */
    BroadcastReceiver mUsbReceiver = new BroadcastReceiver() {
        public void onReceive(Context context, Intent intent) {
            String action = intent.getAction();

            if (UsbManager.ACTION_USB_DEVICE_ATTACHED.
                equals(action)) {
                    mBaudrate = loadDefaultBaudrate();
                    mSerial.begin();
            }
```

Home

```
else if(UsbManager.ACTION_USB_DEVICE_DETACHED.
    equals(action)) {
    mSerial.usbDetached(intent);
    mSerial.end();
}
else if (ACTION_USB_PERMISSION.equals(action))
{
/* A permission response has been received, validate if the
   user has  GRANTED, or DENIED permission */
```

Home

```
            if (!mSerial.isConnected()) {
                        mBaudrate = loadDefaultBaudrate();
                        mSerial.begin();
                        loadDefaultSettingValues();
                        mTvSerial.setTextSize(mTextFontSize);
             }
}
```

```
unregisterReceiver(mUsbReceiver); /* Cancel listener */
```

Read Data

```
private void readDataFromSerial() {
        .
        .
        if(!mSerial.isConnected()) return;

            len = mSerial.read(rbuf);
        .
        .
}
```

Home

Write Data

```
private void writeDataToSerial() {

        if(!mSerial.isConnected()) return;

    String strWrite = etWrite.getText().toString();
    if (SHOW_DEBUG) {
      Log.d(TAG, "PL2303Driver Write(" + strWrite.length() + ") :
            " + strWrite);
    }
    mSerial.write(strWrite.getBytes(), strWrite.length());
  }
```

Home

The Drop-Down Menu Program

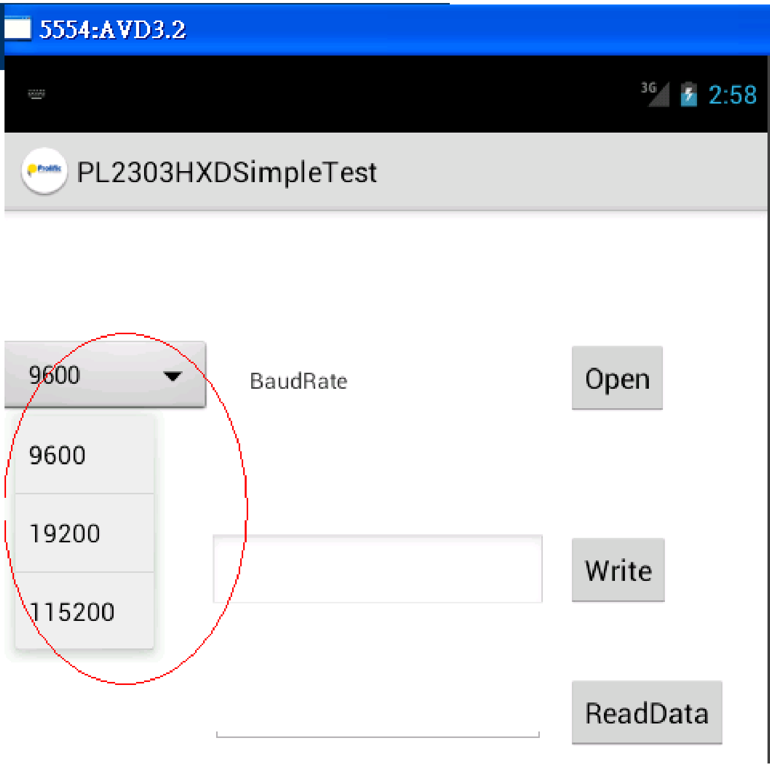
Src\PL2303HXDSimpleTest.java

```
PL2303HXD_BaudRate_spinner =
(Spinner)findViewById(R.id.spinner1);

adapter.setDropDownViewResource
   (android.R.layout.simple_spinner_dropdown_item);

PL2303HXD_BaudRate_spinner.setAdapter(adapter);
```

http://www.myandroid.tw/bbs-topic-26.sea

Res\layout\activity_pl2303_hxdsimple_test.xml

```
<Spinner
    android:id="@+id/spinner1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignTop="@+id/button1"
    android:background=
              "@android:drawable/btn_dropdown" />
```



Home